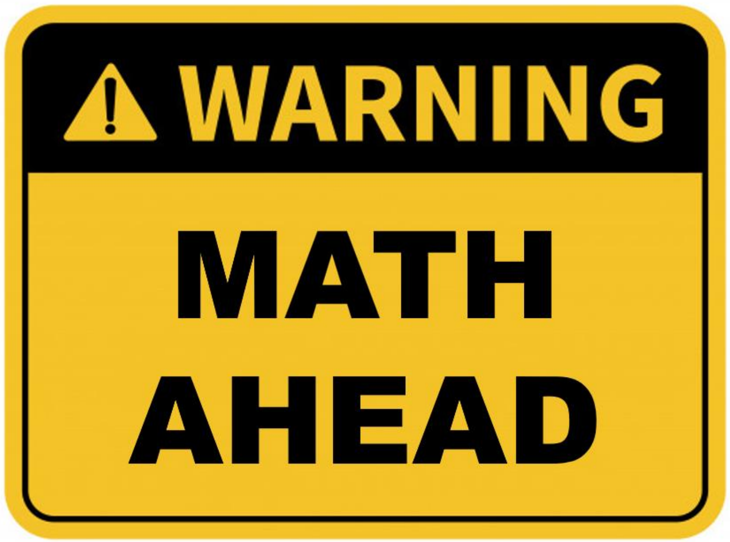
Variational Autoencoders

18-789

Lecture 5



						readings	
1/29/2025	Wednesday	5	Varitional Autoencoders (VAEs)	Lecture 5	3	[Joseph]	
2/3/2025	Monday	6	Student Presentation I: Architecture for Sequence Modeling		4		
2/5/2025	Wednesday	7	Generative Adversarial Networks (GANs)	Lecture 6	4	[Lilian I]	
2/10/2025	Monday	8	Student Presentaiton II: Improving the Stability of Training GANs		5		HW1 due
2/12/2025	Wednesday	9	Normalizing Flows/Invertible Models	Lecture 7	5	[Lilian II]	HW2 release
2/17/2025	Monday	10	Introduction to Diffusion Models	Lecture 8	6	[Angus] [Calvin]	
2/19/2025	Wednesday	11	Guest Lecture: Recent Advances in GANs (Jun-Yan Zhu)		6		
2/24/2025	Monday	12	Continuous-time Diffusion Models	Lecture 9	7	[Yang] [Sander I]	
2/26/2025	Wednesday	13	Project proposal presentation		7		HW2 due
3/3/2025	Monday	Spring Break - No class			8		
3/5/2025	Wednesday	Spring Break - No class			8		
3/10/2025	Monday	14	Diffusion Models and Flow Matching	Lecture 10	9	[Ruiqi]	

Next Monday: Student Presentation

- Rubrics:
  - Structure (3pts)
  - Central message (3pts)
  - Communication (2pts)
  - Give broader context (1pt)
  - On time (1pt)
- Asking questions about papers during office hours is highly encouraged.

Recap: Training Autoregressive Models

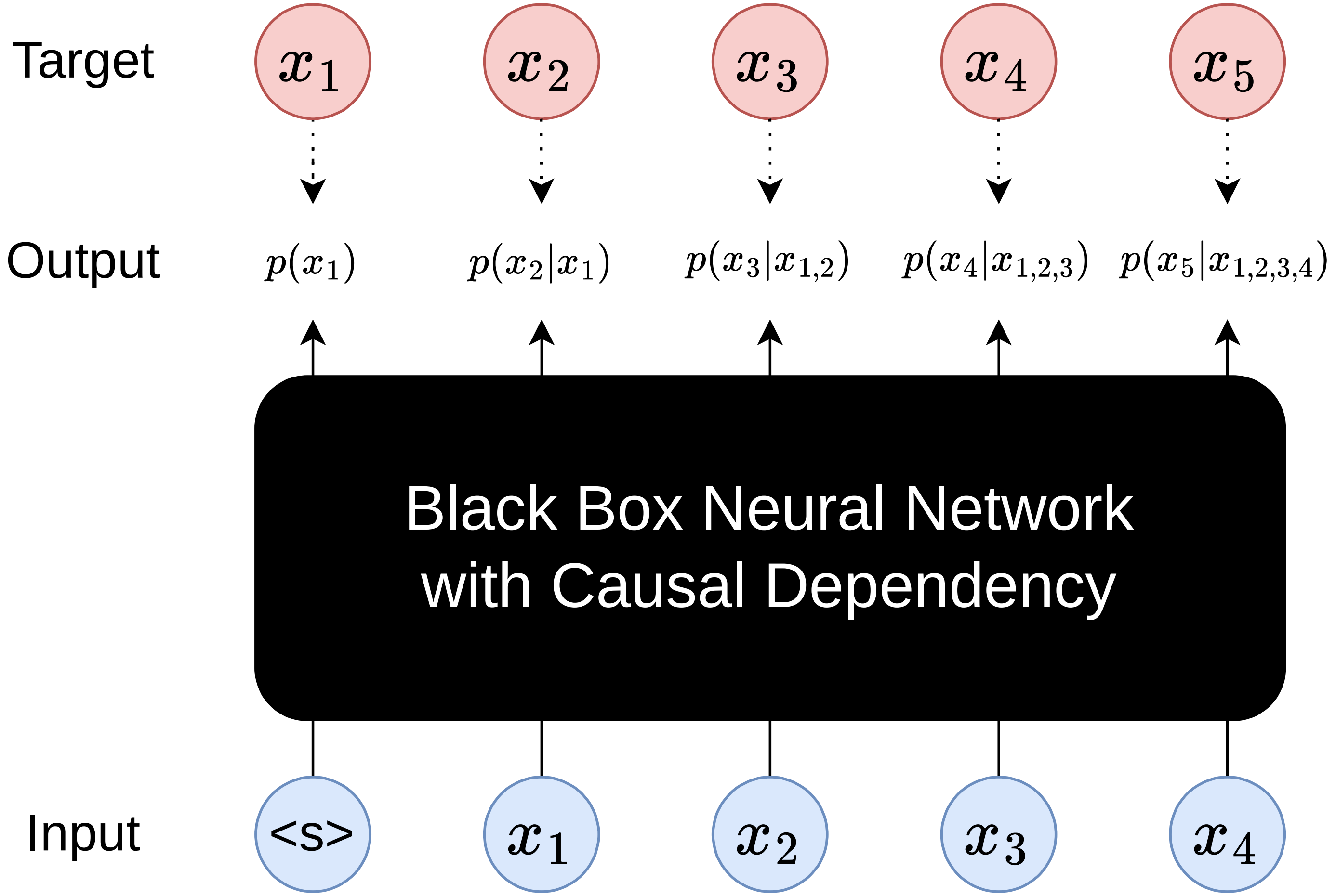
Chain rule decomposition

- $p(x_1, \dots, x_T) = \prod_t p(x_t | x_1, \dots, x_{t-1})$

Maximum Likelihood Estimation (MLE)

- $\underset{\theta}{\operatorname{argmax}} \sum_t \log p_\theta(x_t | x_{1,2,\dots,t-1})$

  (also known as teacher forcing)



Recap: Training Autoregressive Models

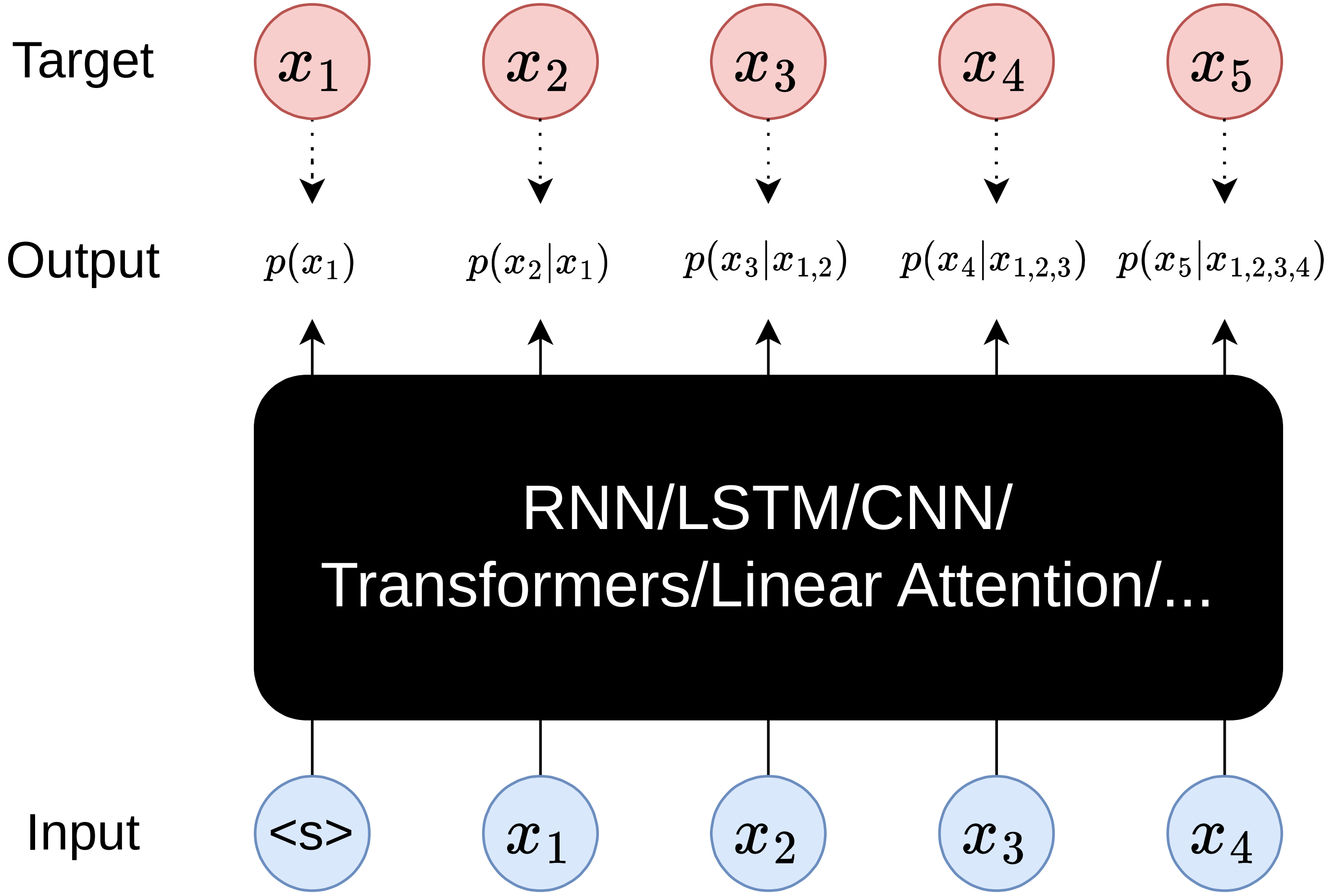
Chain rule decomposition

- $p(x_1, \dots, x_T) = \prod_t p(x_t | x_1, \dots, x_{t-1})$

Maximum Likelihood Estimation (MLE)

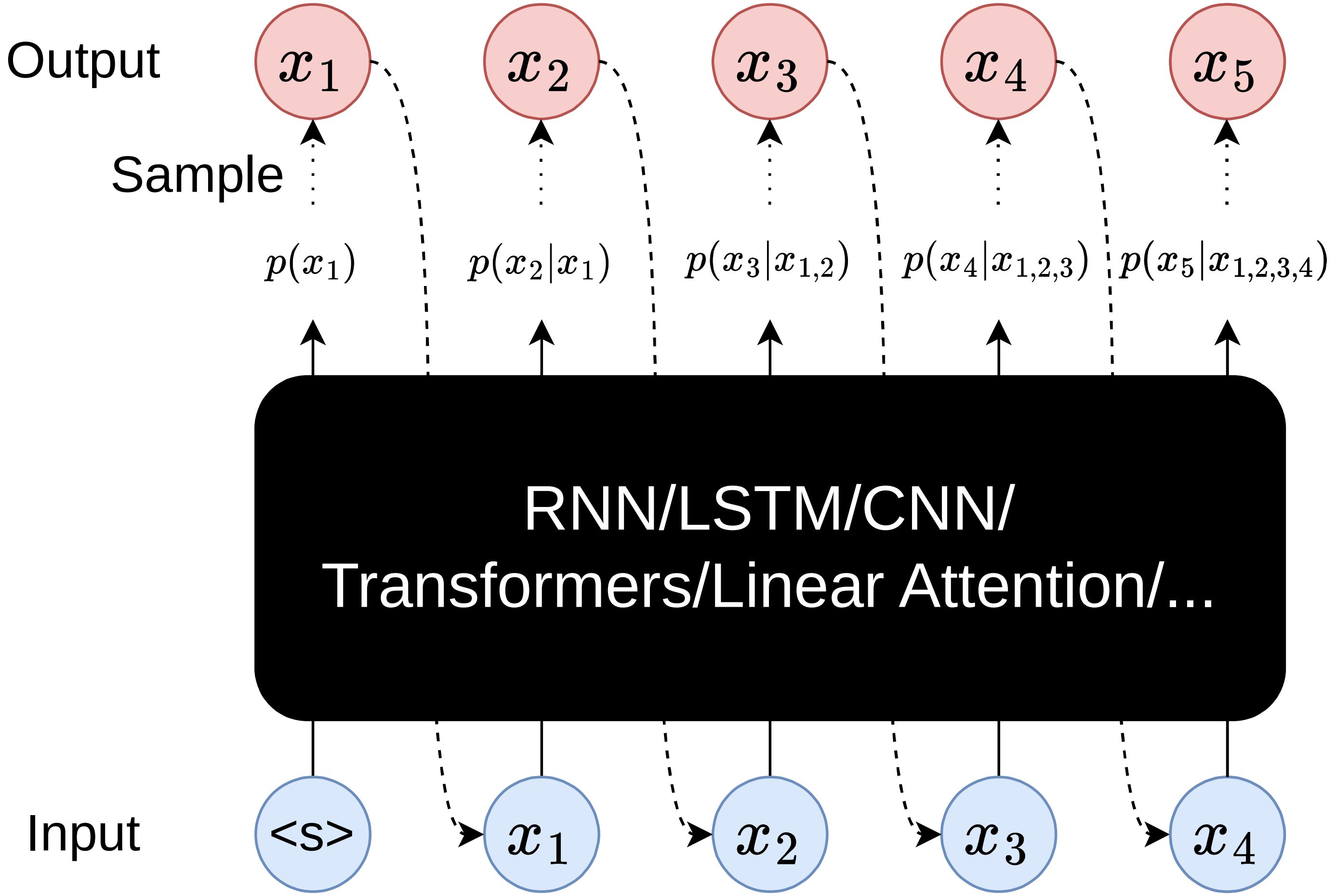
- $\underset{\theta}{\operatorname{argmax}} \sum_t \log p_\theta(x_t | x_{1,2,\dots,t-1})$

  (also known as teacher forcing)



Recap: Sampling from Autoregressive Models

for $t = 1, 2, \ldots, T\ do$:
  sample $x_t \sim p_\theta\left(x_t \middle| x_{1,2,\ldots,t-1}\right)$
return $(x_1, x_2, \ldots, x_T)$



Recap: Autoregressive Models

1. How to model the joint distribution of high-dimensional data?
    - Chain rule decomposition
    - $p_\theta(x_1, x_2, \dots, x_t) = p_\theta(x_1)p_\theta(x_2|x_1) \dots p_\theta(x_t|x_1, x_2, \dots, x_{t-1})$

2. How to optimize your model?
    - Maximizing Likelihood Estimation (MLE)

Autoregressive is not all you need

- Sampling cannot be parallelized
- Error Accumulation
  - Potential train-test distribution mismatch

for $t = 1, 2, \ldots, T\ do$:
    sample $x_t \sim p_\theta\left(x_t \middle| x_{1,2,\ldots,t-1}\right)$
return $(x_1, x_2, \ldots, x_T)$

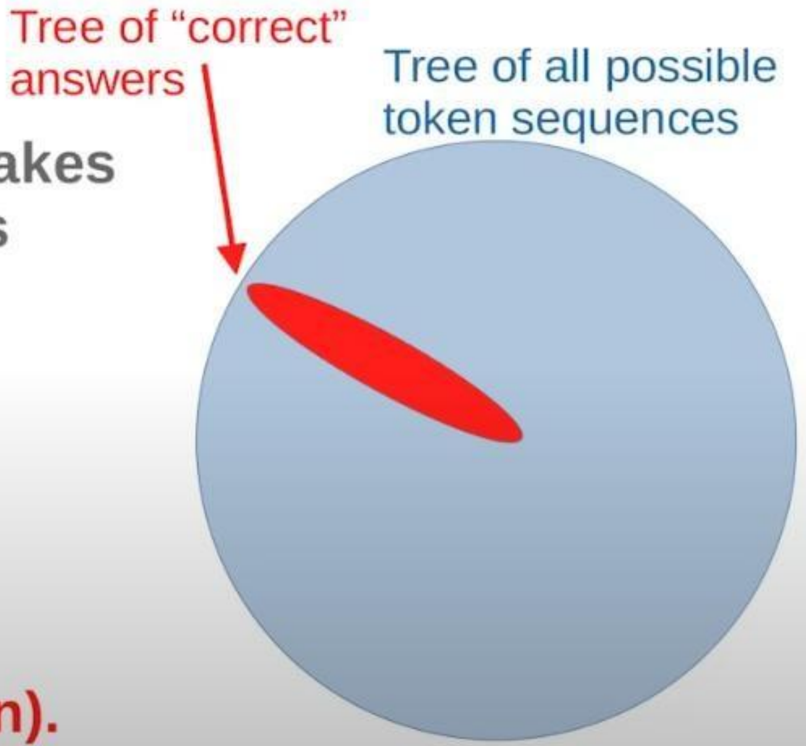
Y. LeCun

Unpopular Opinion about AR-LLMs

- **Auto-Regressive LLMs are doomed.**
- **They cannot be made factual, non-toxic, etc.**
- **They are not controllable**

- **Probability e that any produced token takes us outside of the set of correct answers**
- **Probability that answer of length n is correct:**
  - $P(\text{correct}) = (1-e)^n$

- **This diverges exponentially.**
- **It's not fixable (without a major redesign).**



Autoregressive is not all you need

- Sampling cannot be parallelized

- Error Accumulation
  - Potential train-test distribution mismatch

- Assumes the distribution is roughly time-invariant in some order
  - $p(x_t|x_1, \dots, x_{t-1}) \approx p(x_{t+1}|x_2, \dots, x_t)$

time-invariant

You	love	playing	basketball	,	I	love	playing	tennis

$p(\text{basketball}|\text{love}, \text{playing})$
doesn't change much no matter where it is

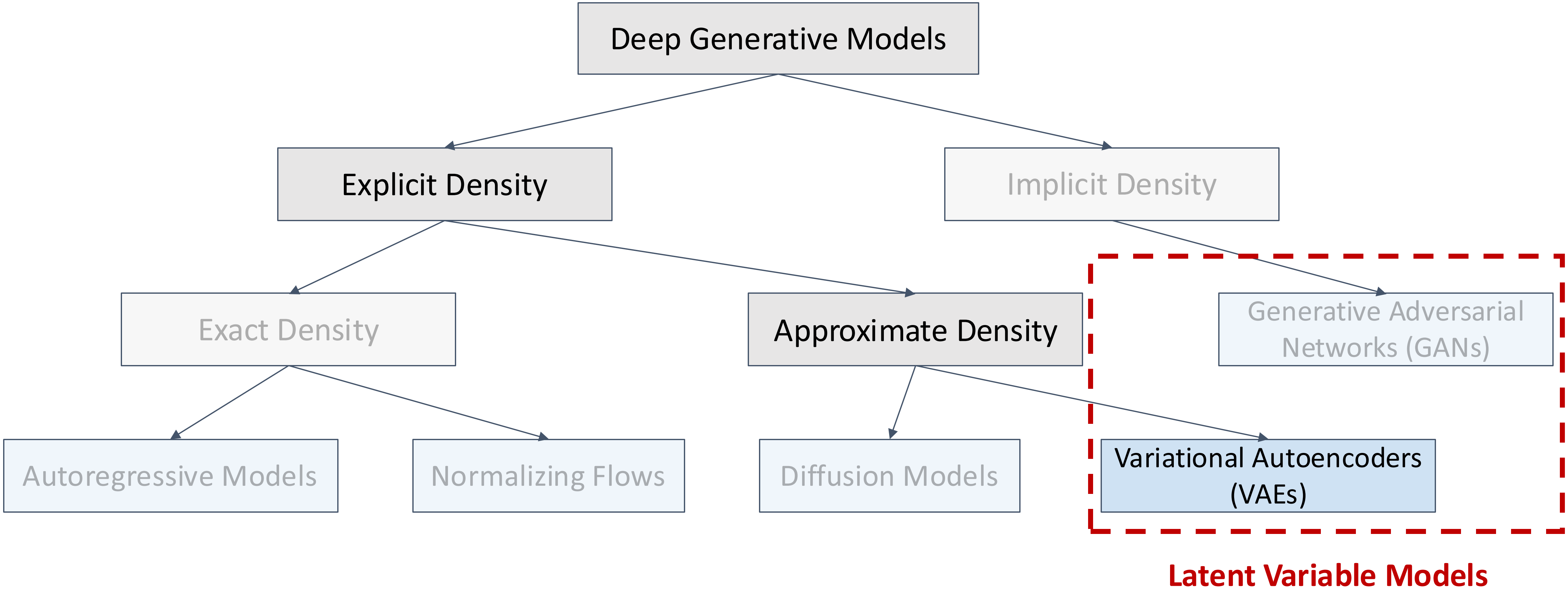
NOT time-invariant

Examples from "Several questions of visual generation in 2024", Gu et al.

Autoregressive is not all you need

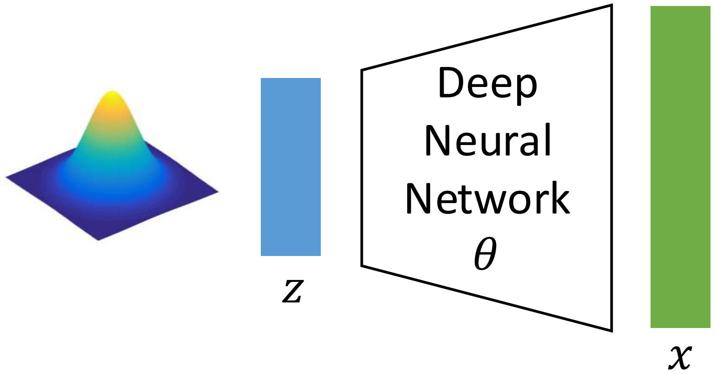
- Sampling cannot be parallelized
- Error Accumulation
  - Potential train-test distribution mismatch
- Assumes the distribution is roughly time-invariant in some order
  - $p(x_t|x_1, \dots, x_{t-1}) \approx p(x_{t+1}|x_2, \dots, x_t)$
- Continuous data?
- Doesn’t learn a compressed representation
  - Compressing the dataset
  - But NOT compressing a data point

Variational Autoencoders



Latent Variable Models

- Data (like images) can be very high-dimensional
- BUT: there exists a latent variable z that is much lower-dimensional and captures important factors of variations.
    - Gender, pose, age, hair color, race, etc.
    - Not observed in our dataset (unless explicitly annotated)
    - $p_\theta(x) = \int_z p_\theta(x, z)\mathrm{dz} = \int_z p_\theta(z)p_\theta(x|z)\mathrm{dz}$
    - Assume $p(z)$ is a simple distribution, e.g., $N(0, I)$ that *hopefully* captures meaningful information
    - Assume $p_\theta(x|z)$ is also simple, e.g., independent Gaussian



Latent Variable Models

1. How to model the joint distribution of high-dimensional data?
    - $p_\theta(x) = \int_z p(z)p_\theta(x|z)\mathrm{dz}$, where z is lower-dimensional
    - $p(z)$ and $p_\theta(x|z)$ are simple distributions that can be *factorized*
        - $p_\theta(\boldsymbol{x}|z) = p_\theta(x^1, x^2, \ldots, x^D|z) = \prod_i p_\theta(x^i|z)$

2. How to optimize your model?

Latent Variable Models

1. How to model the joint distribution of high-dimensional data?
   - $p_\theta(x) = \int_z p(z)p_\theta(x|z)\mathrm{dz}$, where z is lower-dimensional
   - $p(z)$ and $p_\theta(x|z)$ are simple distributions that can be *factorized*
     - $p_\theta(\boldsymbol{x}|z) = p_\theta(x^1, x^2, \ldots, x^D|z) = \prod_i p_\theta(x^i|z)$

2. How to optimize your model?

Maximum Likelihood Estimation (?)

- $\log p_\theta(x) =$

- BUT: $p_\theta(x|z)$ is almost zero for most z
- Why is it a problem?

$$p_\theta(x|z)\begin{cases}\neq 0, \text{if } z = (\text{black}, \text{male}, \text{middle} - \text{age}, \text{frontface}, \dots) \\ = 0, \text{otherwise.}\end{cases}$$

Evidence Lower Bound (ELBO)

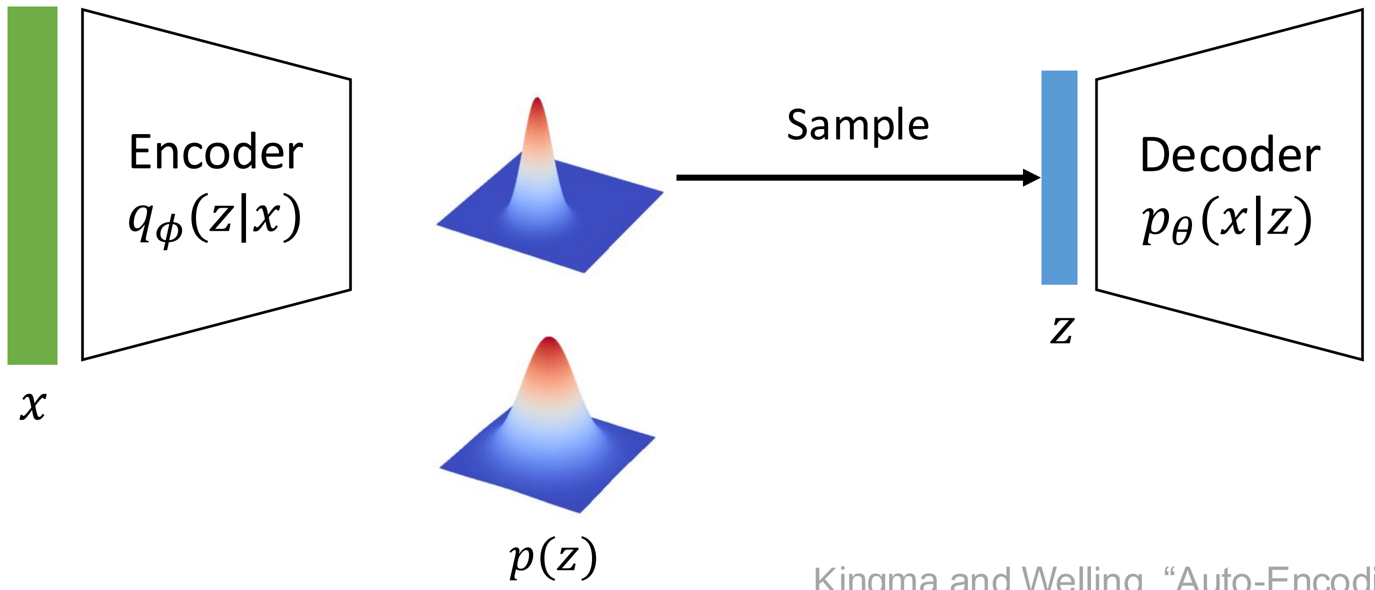
We use a neural network ϕ to predict the "posterior" distribution $q_\phi(z|x)$

Then: $\log p_\theta(x)$

$=$

Variational Autoencoders (VAEs)

$$\min_{\theta,\phi} -\mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] + KL(q_\phi(z|x) \parallel p(z))$$



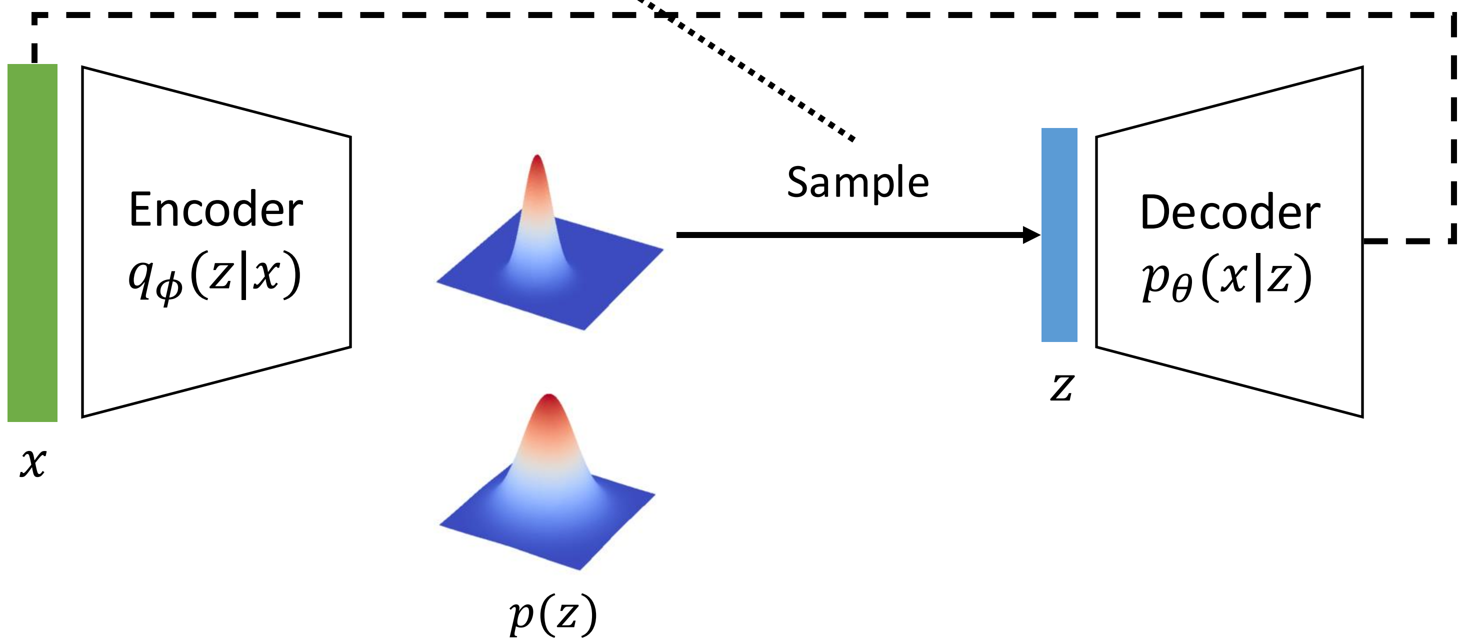
Kingma and Welling, “Auto-Encoding Variational Bayes”, NeurIPS 2014

Reconstruction Loss

$$\min_{\theta,\phi} -\mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] + KL(q_\phi(z|x) \parallel p(z))$$

- How to evaluate $\log p_\theta(x|z)$?
- Idea: Gaussian likelihood decoder $p_\theta(x|z) = N(x|\mu_\theta(z), \sigma^2)$

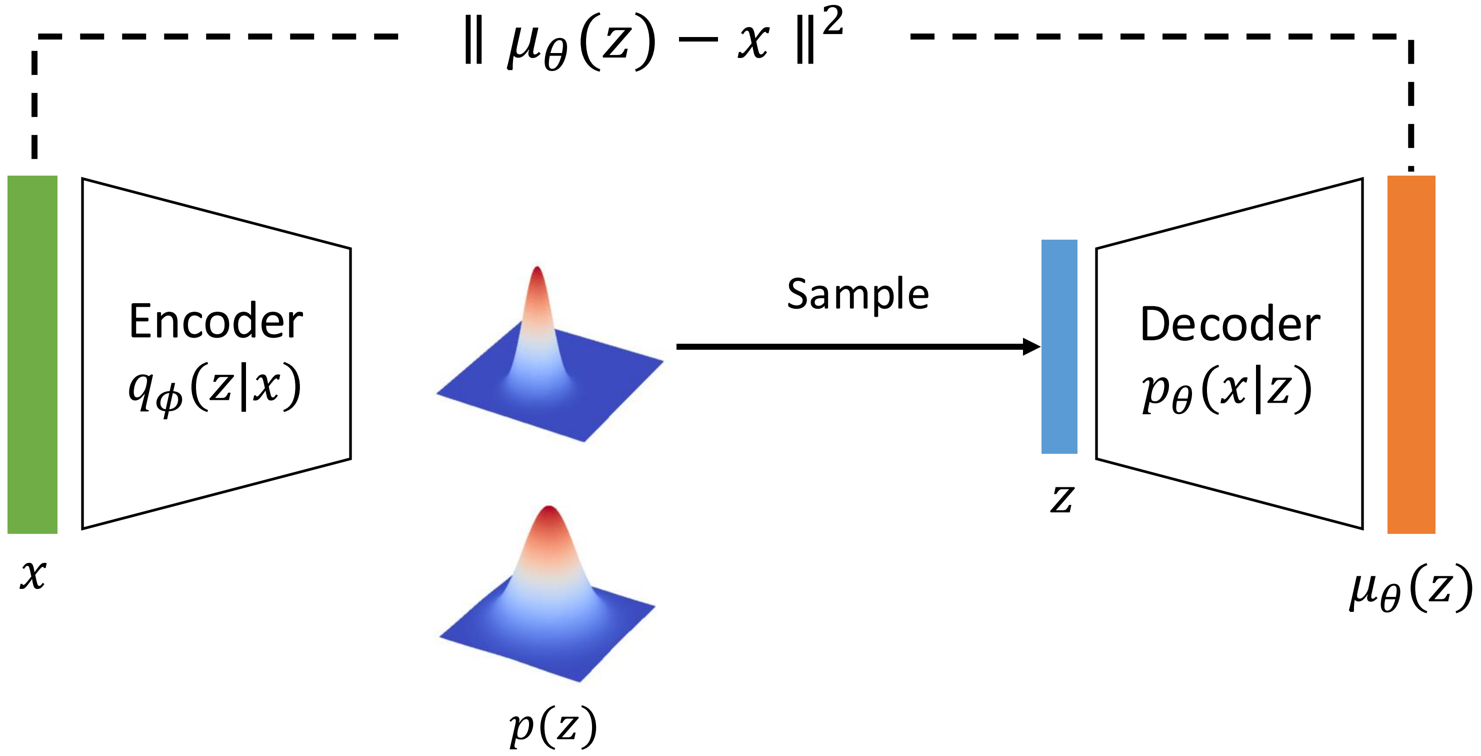
Monte Carlo Estimation



Reconstruction Loss

$$\min_{\theta,\phi} -\mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] + KL(q_\phi(z|x) \parallel p(z))$$

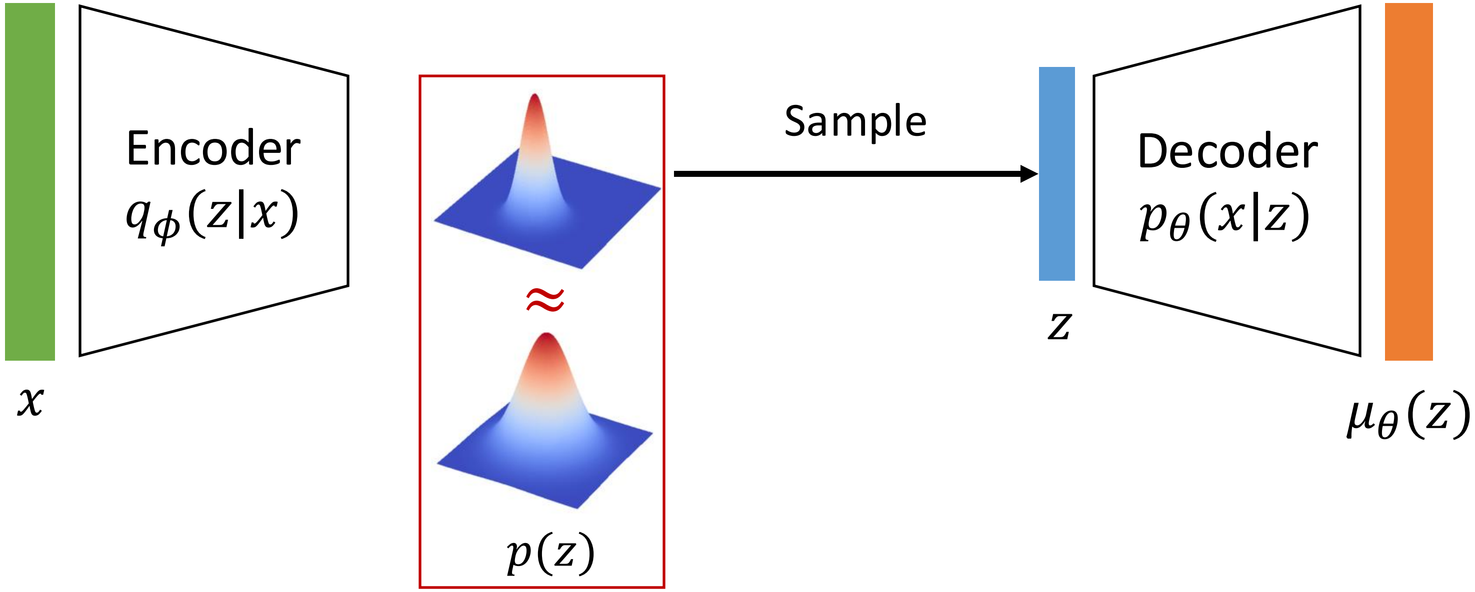
$$=$$



Regularization Loss (KL Loss)

$$\min_{\theta,\phi} \mathbb{E}_{z\sim q_\phi(z|x)} \| \mu_\theta(z) - x \|^2 + KL(q_\phi(z|x) \| p(z))$$

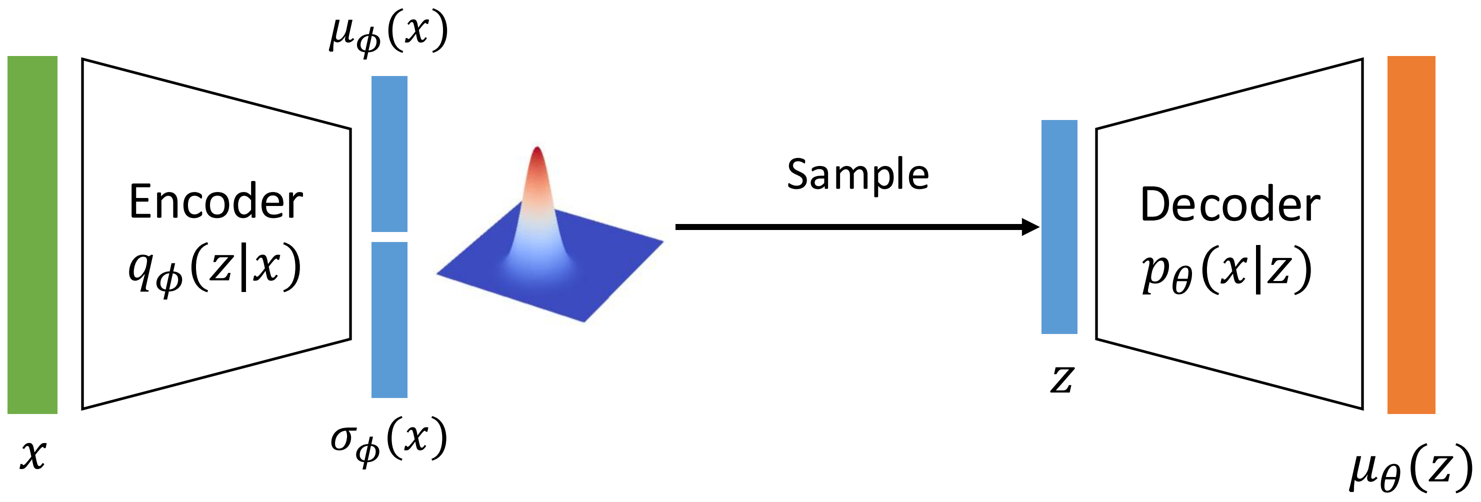
- How to evaluate the KL divergence?
- Idea: Gaussian posterior $q_\phi(z|x) = N(\mu_\phi(x), \sigma_\phi^2(x))$ because the KL divergence between Gaussians can be computed in closed form.



Reparameterization Trick

$$\min_{\theta,\phi} \mathbb{E}_{z\sim q_\phi(z|x)} \| \mu_\theta(z) - x \|^2 + KL(q_\phi(z|x) \| p(z))$$

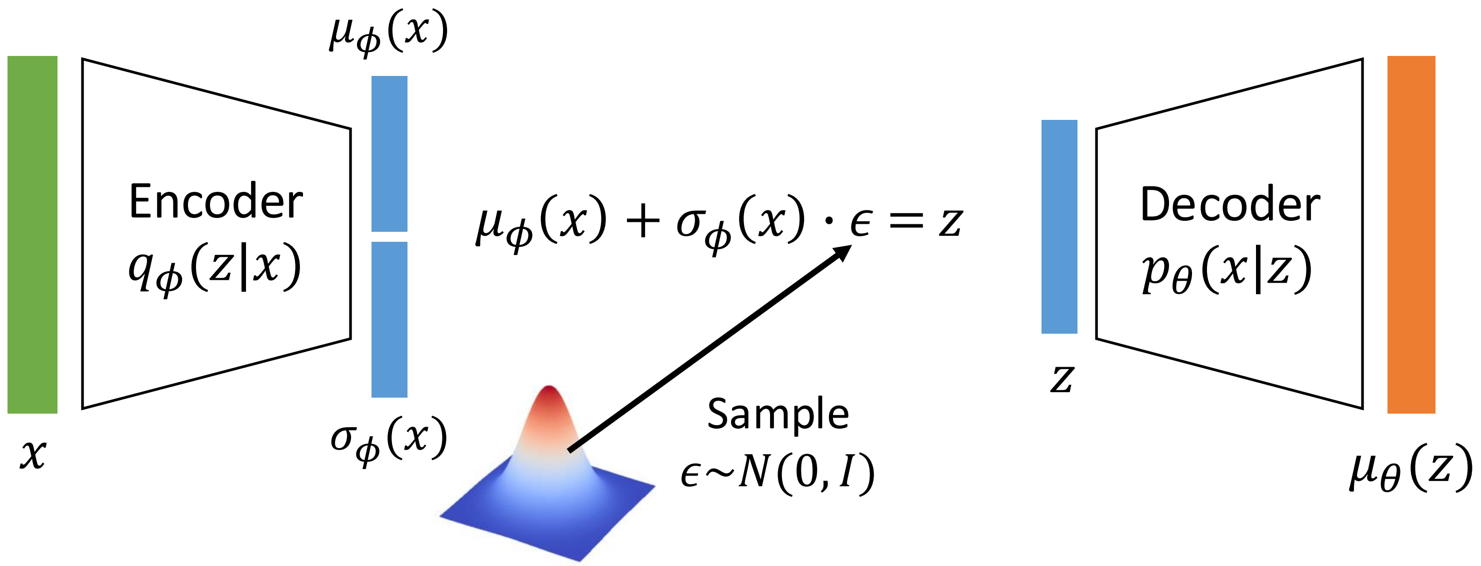
- To train neural networks with backpropagation, the computation needs to be differentiable. But sampling from a distribution is not differentiable.



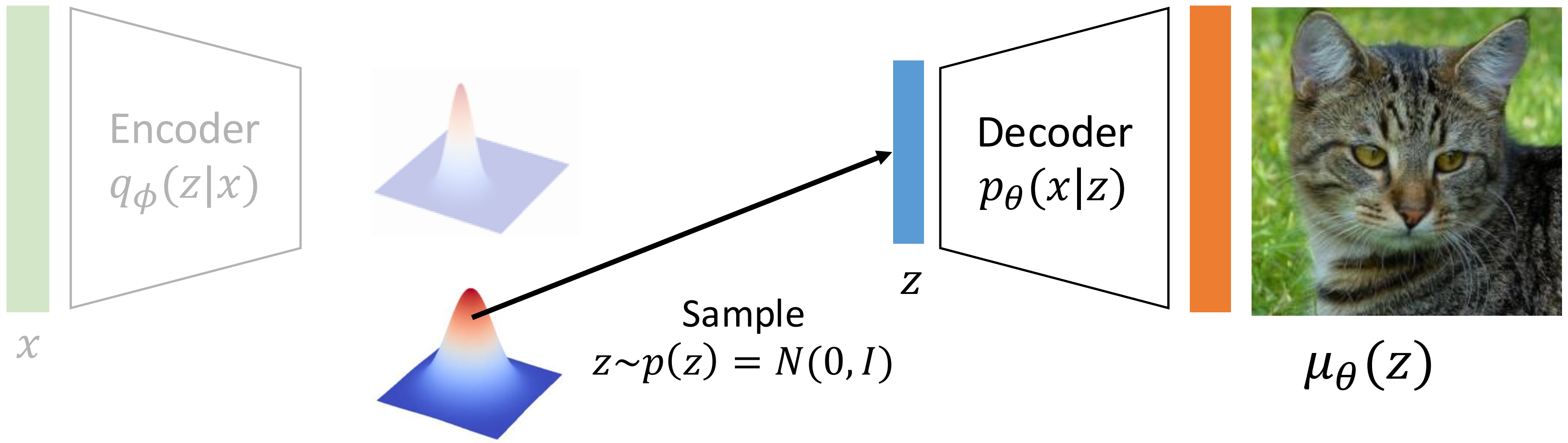
Reparameterization Trick

$$\min_{\theta,\phi} \mathbb{E}_{z\sim q_\phi(z|x)} \parallel \mu_\theta(z) - x \parallel^2 + KL(q_\phi(z|x) \parallel p(z))$$

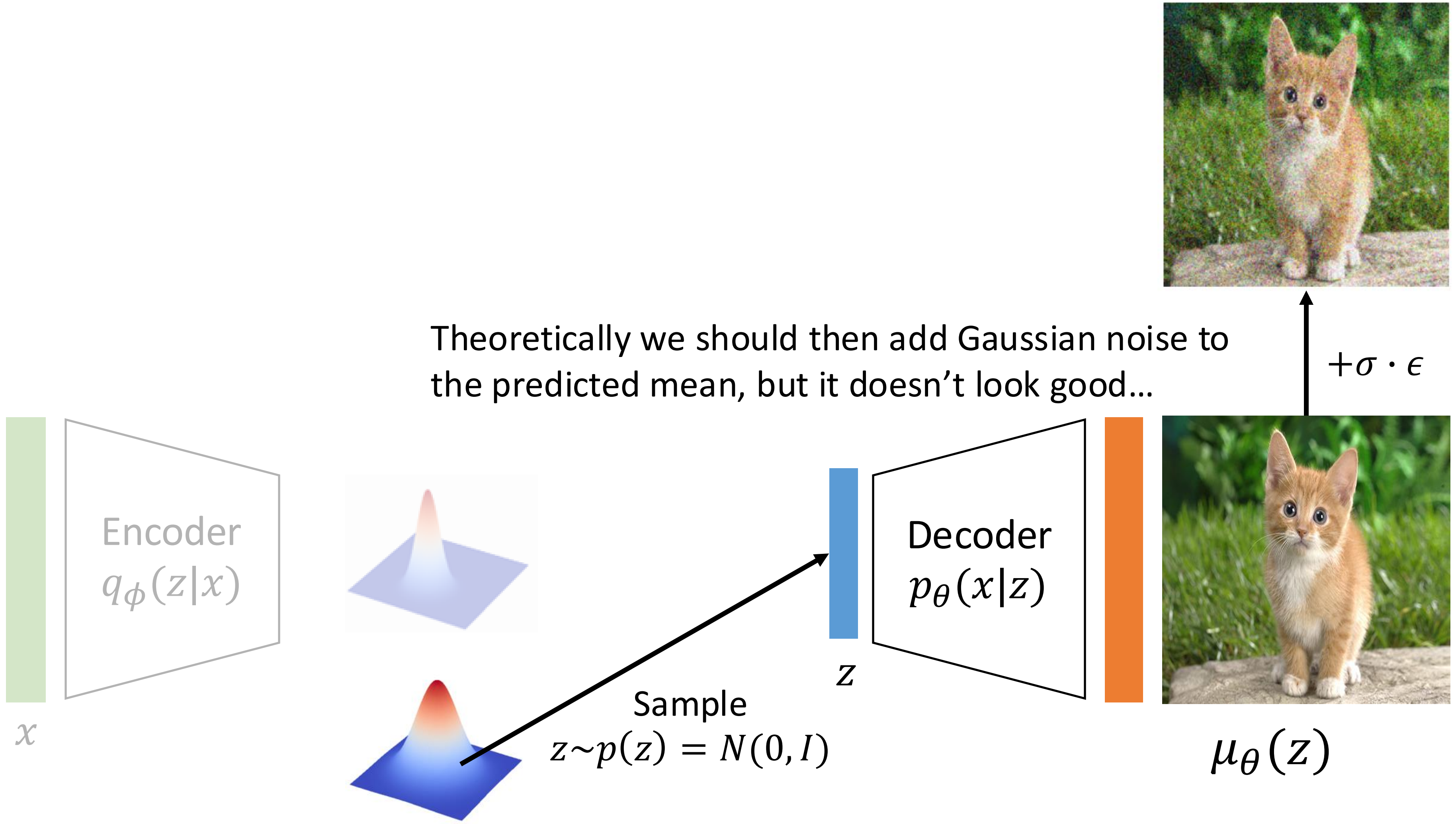
- To train neural networks with backpropagation, the computation needs to be differentiable. But sampling from a distribution is not differentiable.



Sampling from a Variational Autoencoder

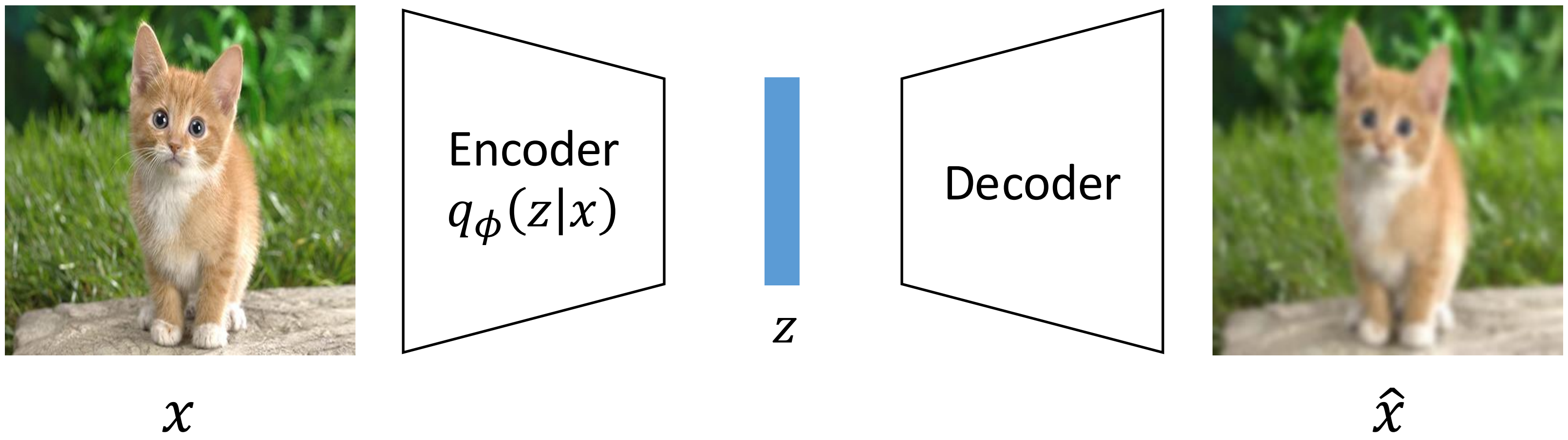


Sampling from a Variational Autoencoder



Intuitive interpretation: Regularized Autoencoder

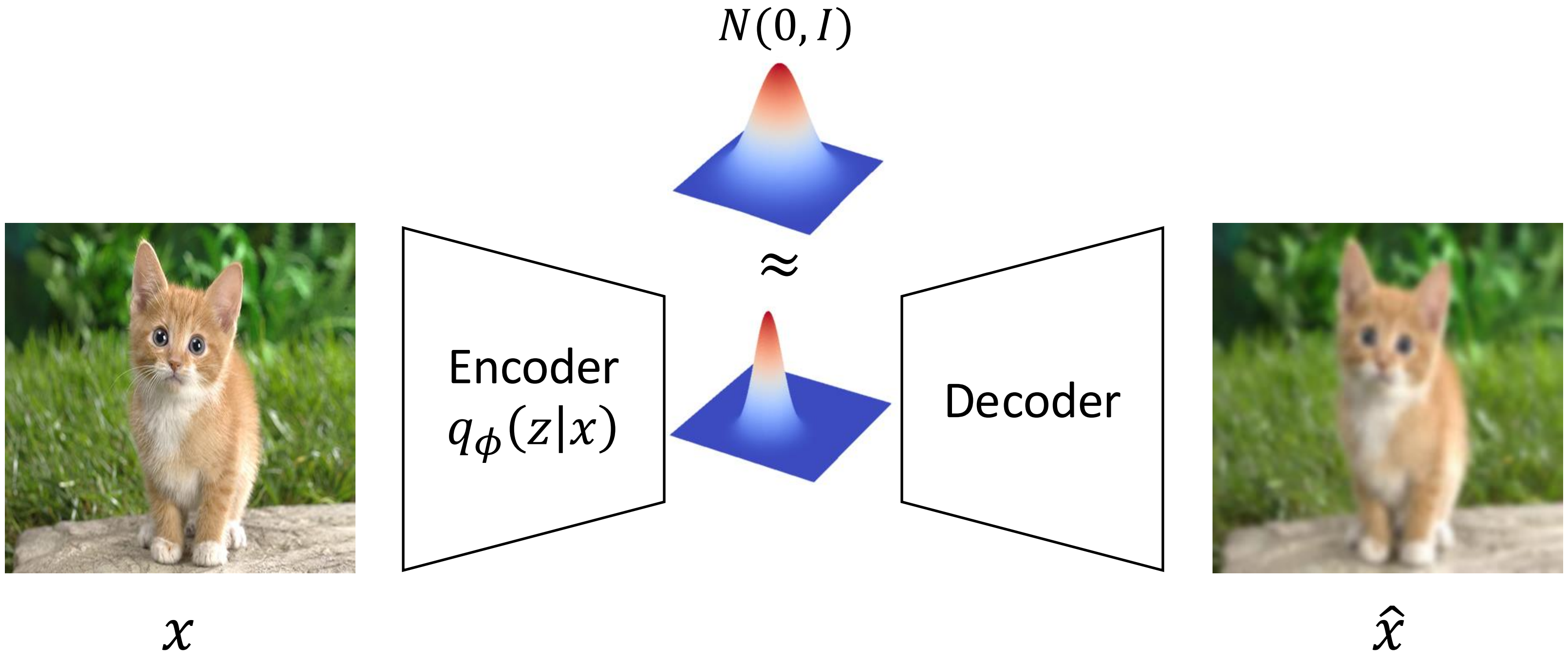
- **Autoencoder** is a neural network that’s trained to reconstruct data while passing through a low-dimensional bottleneck
- But it cannot generate data because the latent distribution is unknown



$$L = \| \hat{x} - x \|^2$$

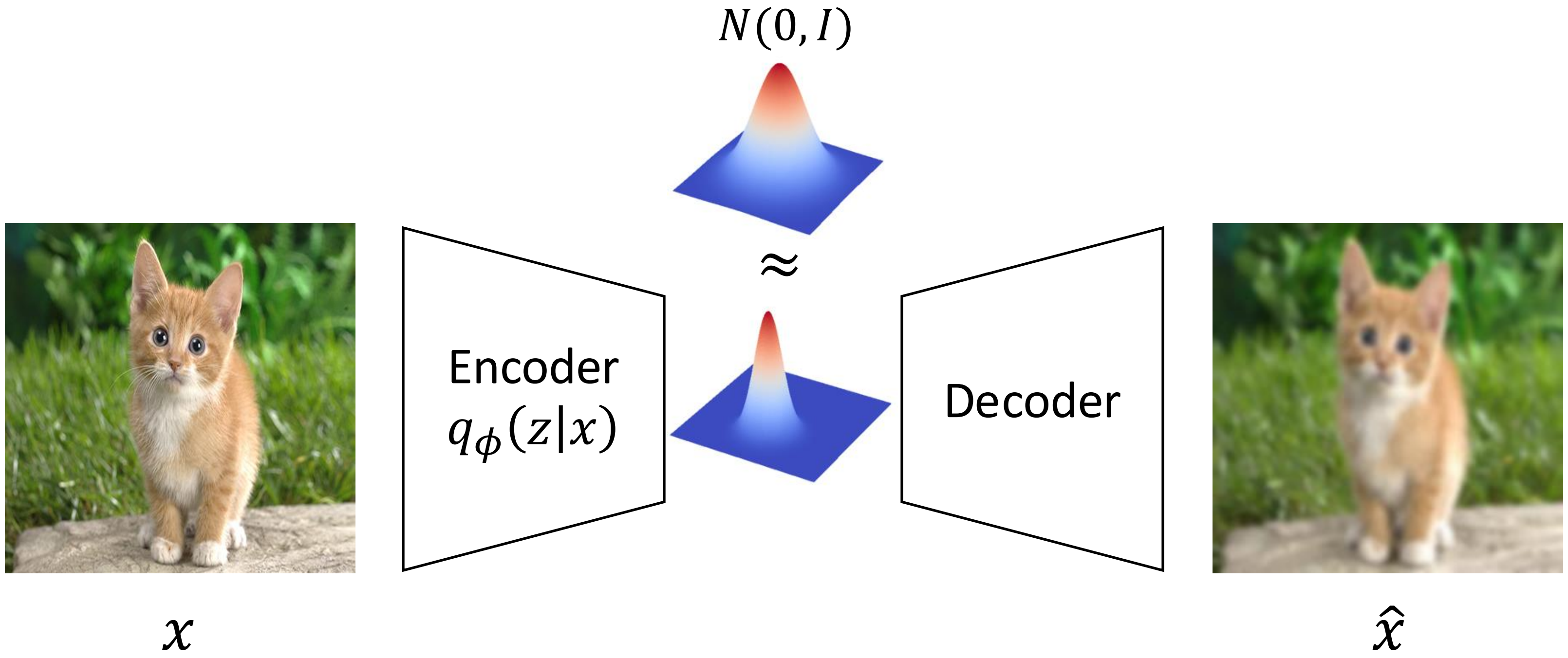
Intuitive interpretation: Regularized Autoencoder

- Add a regularization so that the latent distribution is close to a Gaussian
- So we can sample latent from Gaussian and generate new samples



$$L = \| \hat{x} - x \|^2 + KL(q_\phi(z|x) \parallel p(z))$$

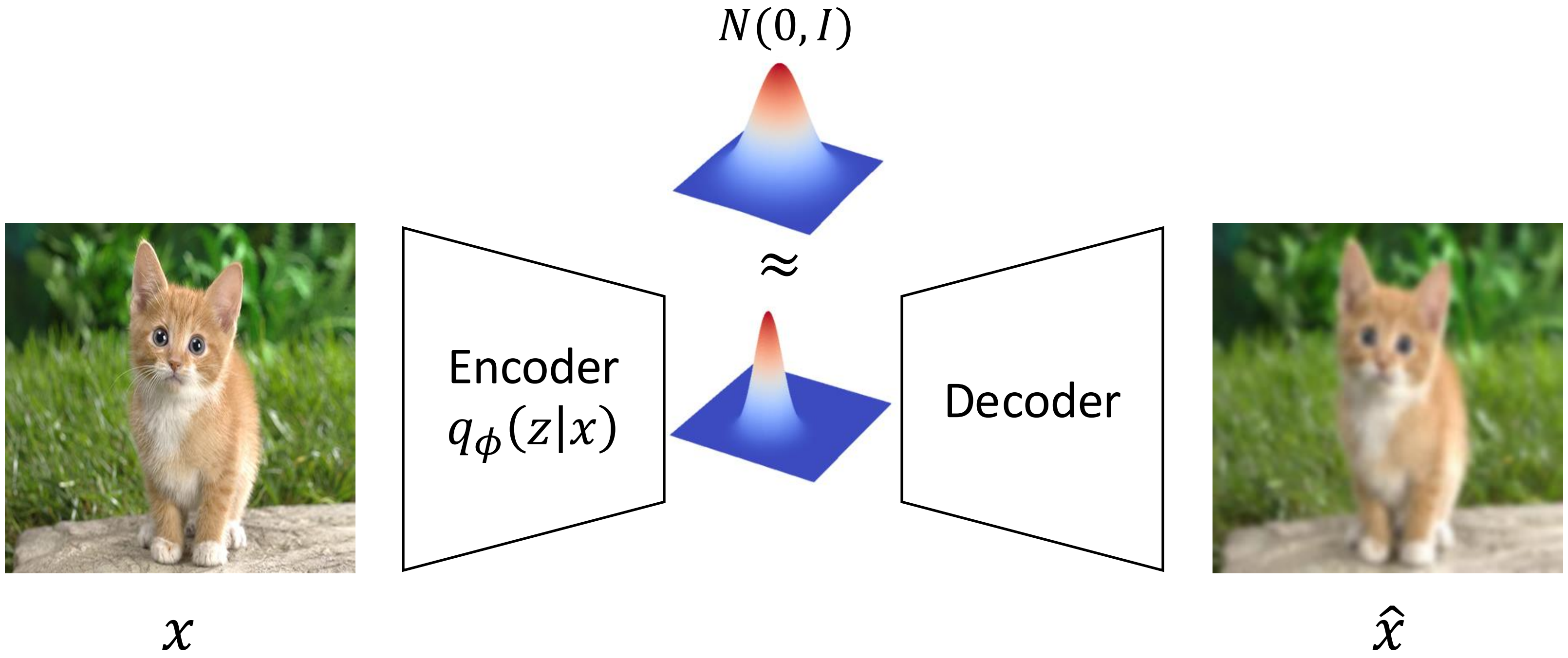
Beta-VAE: adjusting the strength of regularization



$$L = \frac{1}{2\sigma^2} \parallel \hat{x} - x \parallel^2 + KL(q_\phi(z|x) \parallel p(z))$$

Beta-VAE: adjusting the strength of regularization

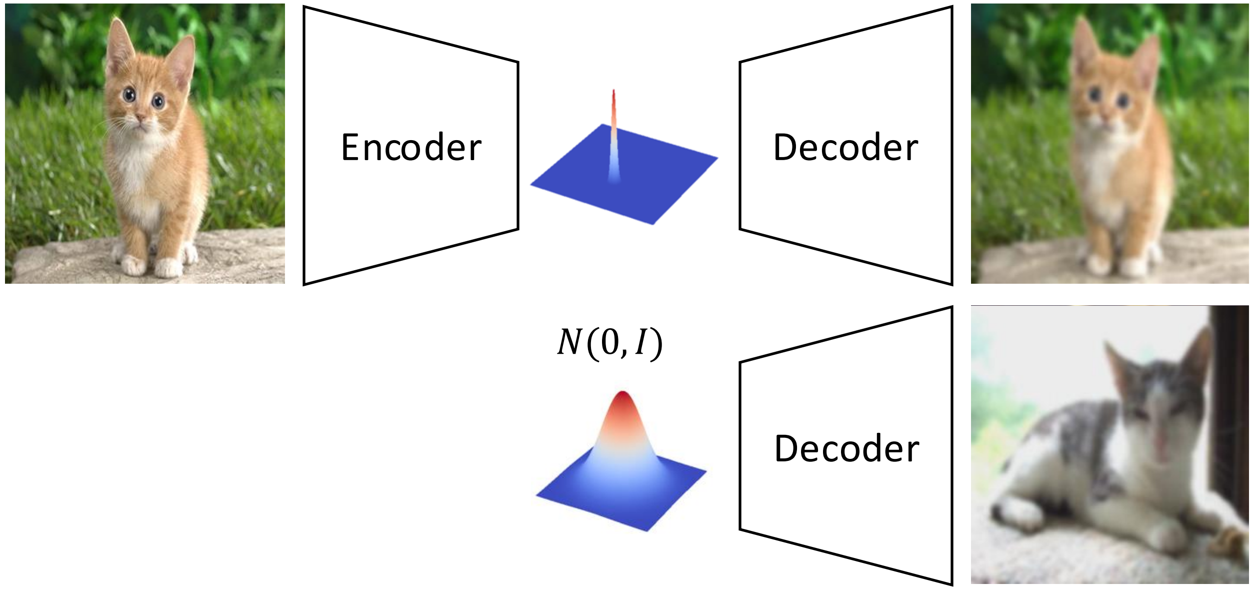
- We can adjust the strength of regularization by increasing the weight of KL loss.



$$L \propto \| \hat{x} - x \|^2 + \beta \, KL(q_\phi(z|x) \parallel p(z))$$

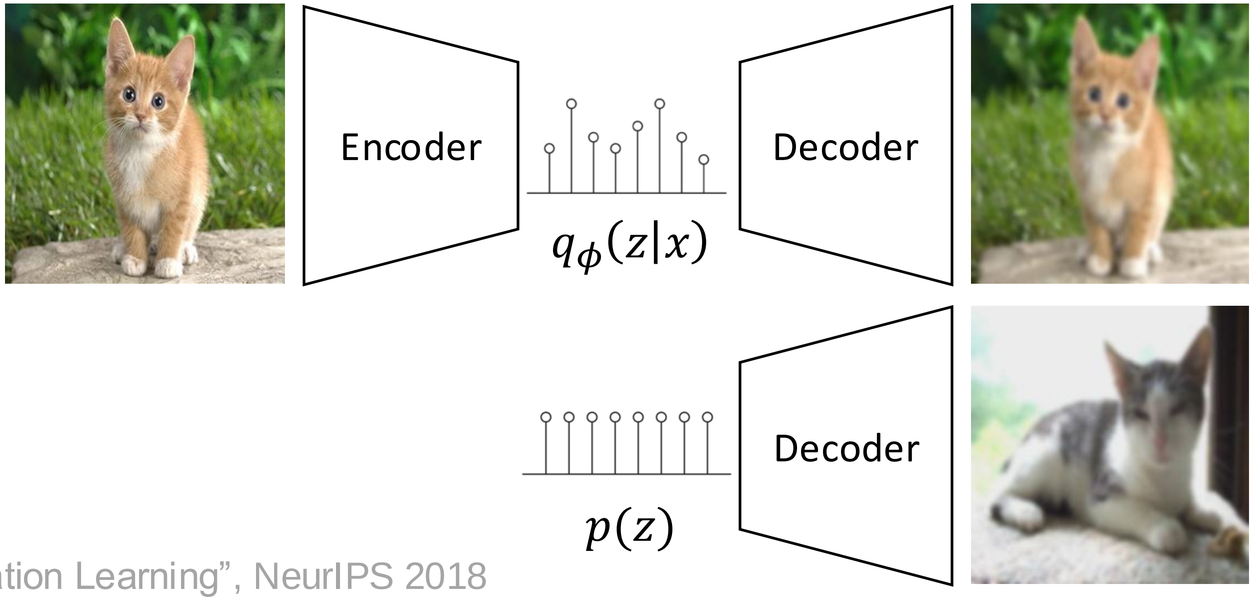
Reconstruction vs. generation tradeoff

- $L = \| \hat{x} - x \|^2 + \beta \; KL(q_\phi(z|x) \parallel p(z))$
- When β is too small: good reconstruction, bad generation
- Increasing β
  - Reconstruction becomes worse
  - Generated images become more like reconstructed images



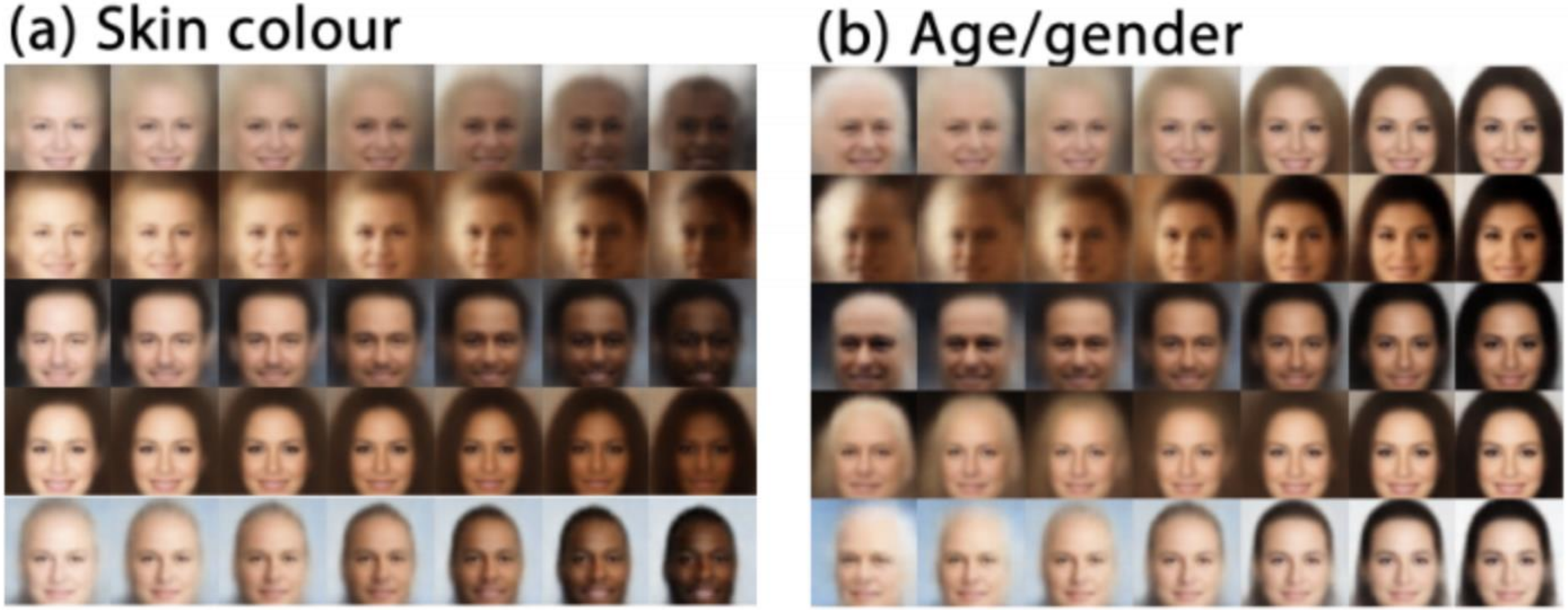
VQ-VAE: VAE with Discrete Latent

- $L = \| \hat{x} - x \|^2 + \beta\, KL(q_\phi(z|x) \,\|\, p(z))$
- Will revisit it in a student presentation later
  - Taming Transformers for High-Resolution Image Synthesis



Oord et al., "Neural Discrete Representation Learning", NeurIPS 2018

VAEs learn semantic latent space without supervision



Higgins et al., “beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework”, ICLR 2017

VAE vs. Autoregressive Models

- Sampling cannot be parallelized
- Error Accumulation
- Assumes the distribution is roughly time-invariant in some order
- Discrete Data only
- Doesn’t learn a compressed representation

VAE vs. Autoregressive Models

- Sampling ~~cannot be~~ parallelized
- ~~Error Accumulation~~
- ~~Assumes the distribution is roughly time-invariant in some order~~
- Discrete Data only
- Doesn’t learn a compressed representation

No
Sequential Order!

VAE vs. Autoregressive Models

- Sampling ~~cannot be~~ parallelized
- ~~Error Accumulation~~
- ~~Assumes the distribution is roughly time-invariant in some order~~
- ~~Discrete Data only~~
- Doesn’t learn a compressed representation

Works on
Continuous Data!

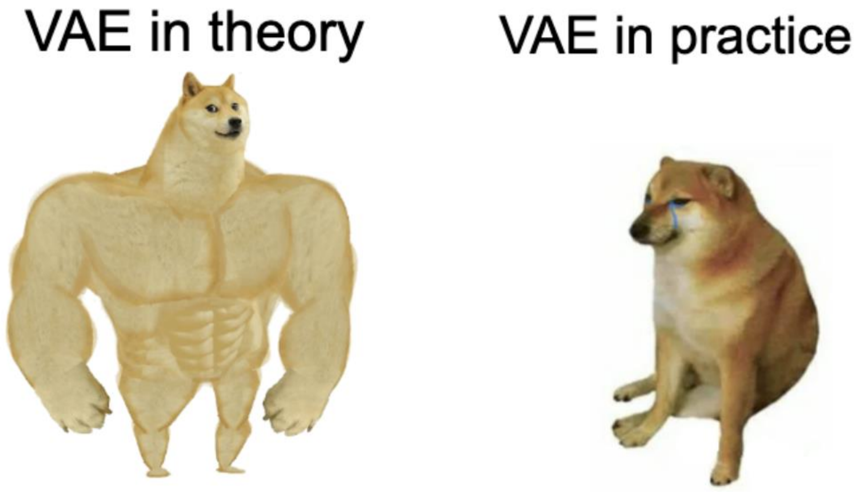
VAE vs. Autoregressive Models

- Sampling ~~cannot be~~ parallelized
- ~~Error Accumulation~~
- ~~Assumes the distribution is roughly time-invariant in some order~~
- ~~Discrete Data only~~
- ~~Doesn’t learn a compressed representation~~

Have
Latent Space!

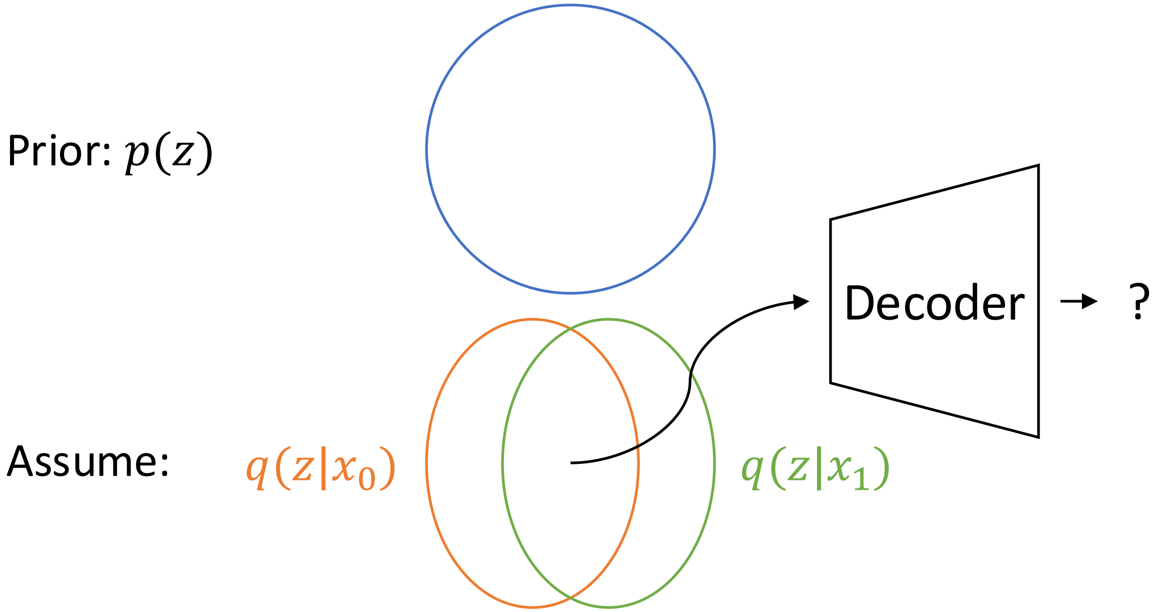
How do VAEs perform?

The output images are blurry!



Why are output images from VAEs blurry?

- Assume our dataset only has two samples: $\{x_0, x_1\}$.
- With optimized reconstruction loss, what would the decoder output if we sample from the origin?



$$\frac{x_0 + x_1}{2}$$

Why are output images from VAEs blurry?

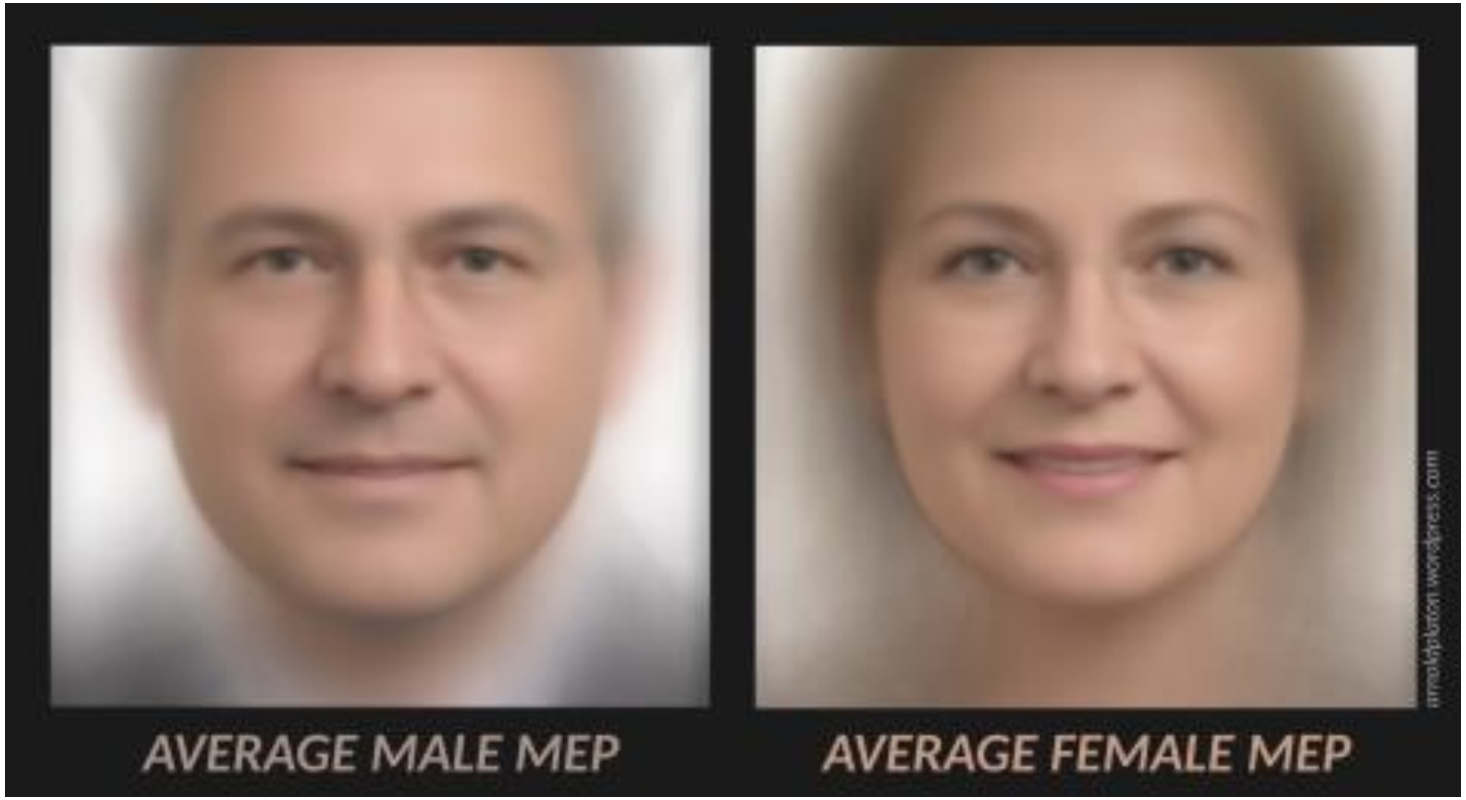
- The optimal decoder output given latent z' is a weighted combination of samples in the training set $\{x_i\}$:

$$\mu_\theta(z') = \sum_i w_i x_i$$

where $w_i = \frac{q(z'|x_i)}{\sum_i q(z'|x_i)}$

The average of many images is blurry

- Average face of European parliament



Why are output images from VAEs blurry?

- The optimal decoder output given latent z' is a weighted combination of samples in the training set $\{x_i\}$:

$$\mu_\theta(z') = \sum_i w_i x_i$$

where $w_i = \frac{q(z'|x_i)}{\sum_i q(z'|x_i)}$

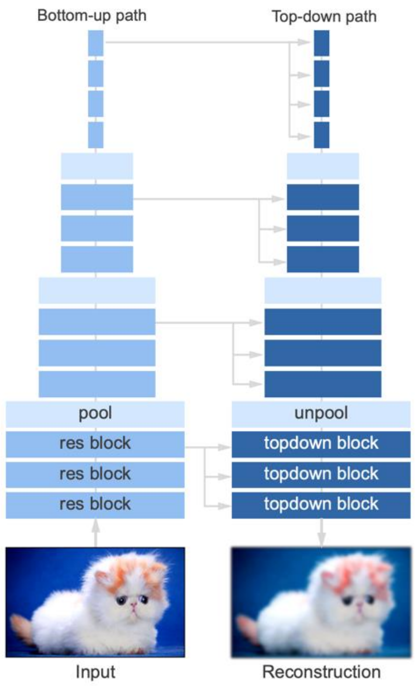
- Another answer: VAE outputs are blurry because of Gaussian assumptions
  - Gaussian likelihood -> The optimal decoder output is a weighted average of training samples.
  - Gaussian posterior + prior -> There is always overlap between posterior distributions, so weights are never one-hot.

VAE: The Curse of Blurriness

- Blurriness is a fundamental problem of VAEs that can't be easily solved by scaling up

256x256

1024x1024



Very Deep (72 layers) +
Hierarchical Latent Space

Child, Very deep vaes generalize autoregressive models and can outperform them on images, ICLR 2020

Non-Gaussian Decoder?

- $\min_{\theta,\phi} -\mathbb{E}_{z\sim q_\phi(z|x)}[\log \underbrace{p_\theta(x|z)}_{\text{PixelCNN}}] + KL(q_\phi(z|x) \parallel p(z))$
- Can we use an autoregressive decoder (e.g. PixelCNN)?
- Posterior Collapse!
  - Encoder ignores x ($q_\phi(z|x) \equiv p(z)$ regardless of x)
  - Decoder ignores z ($p_\theta(x|z) \equiv p_\theta(x)$ regardless of z)
- Degenerate to an autoregressive model…

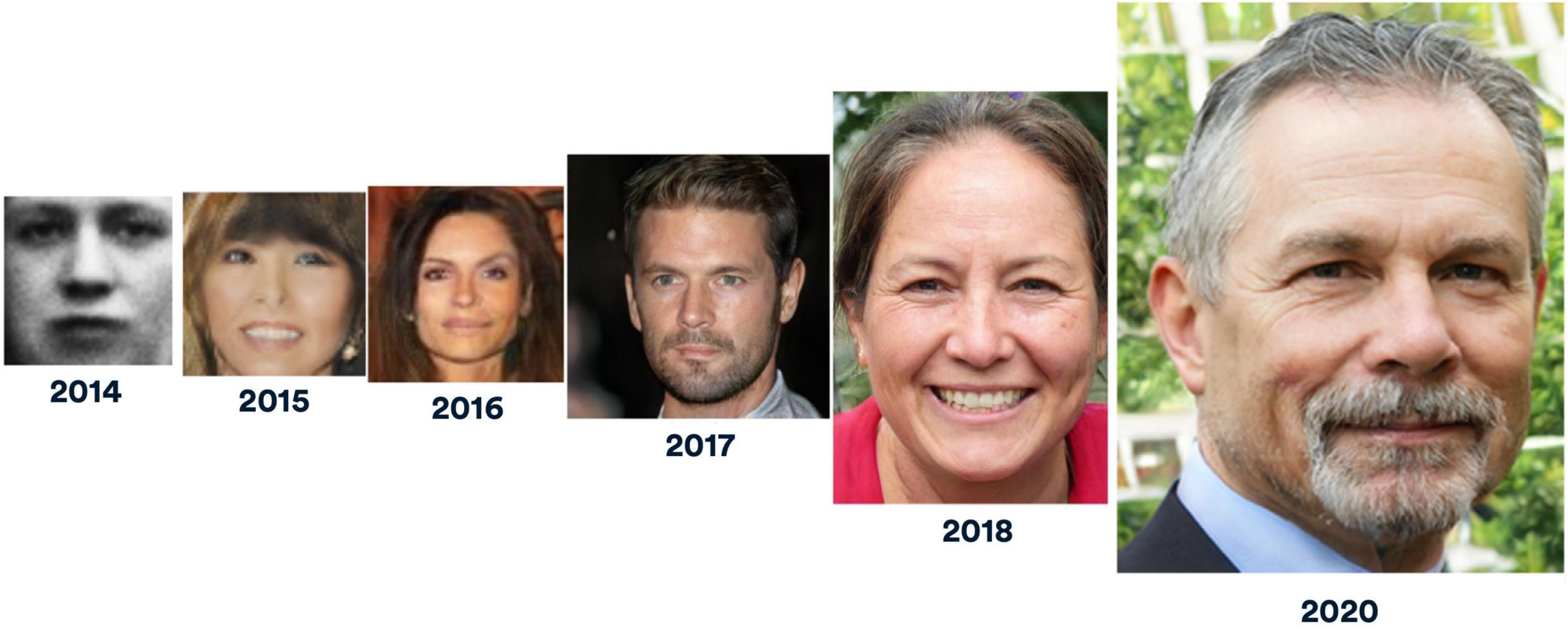
Chen et al., “Variational lossy autoencoder”, ICLR 2017

Recap

- VAEs maximize ELBO, a lower bound of the log-likelihood
  - $\mathbb{E}_{z \sim q_\phi(z|x)}[\log p_\theta(x|z)] - KL(q_\phi(z|x) \parallel p(z))$
- Maximizing Gaussian $\log p_\theta(x|z)$ = Minimizing MSE/L2 loss
- Assume Gaussian $q_\phi(z|x)$ and $p(z)$ so we can compute their KL analytically
- Reparameterization trick to enable gradient backpropagation
- Autoencoder perspective, beta-VAE and VQVAE
- Why VAE outputs blurry images.
  - Gaussian assumptions are too simplistic ☹

Next time: Generative Adversarial Networks (GANs)

- The first generative models that can generate sharp images.



Goodfellow et al., 2014; Radford et al., 2016; Liu & Tuzel, 2016; Karras et al., 2018; Karras et al., 2019; Goodfellow, 2019; Karras et al., 2020; AI Index, 2021

Quiz – On Canvas

- Code: Snake